

# Taxonomy for Choosing BI Systems into an Existing Infrastructure

R. Galli

Széchenyi István University, Department of Information Sciences  
H-9026, Győr, Egyetem tér 1. Hungary  
richard@sze.hu

**Abstract** Business intelligence, as a phrase was first used in 1989. The main reason for this was, that the commercial and sales experts at software development companies felt this expression could be better sold, than the used „decision support system”. Since, there is no practical border between the two expressions. There is only a minor difference in adaptability: is the problem structured enough? Decades ago, it had to be, if the company wanted to use a decision support system, but today, it is not necessary the condition: At the business intelligence systems, there are many ways to solve a problem, the focus is always on the analysis, and thanks to the technological development, serious automation is possible, with more detailed decision support.

*Keywords: business intelligence, business processes, software, platform, infrastructure*

## 1. Introduction

In many occasions appears the following situation at logistics service providers as well: a new business intelligence system needs to be connected to the already existing (and running) infrastructure. On the other hand it would be helpful, if the connection (the system introduction) procedure would not affect the systems and business processes. So it would be ideal to have a business intelligence taxonomy, which could help the experts choose those solutions worth considering, and those not worth considering. The examination will process the systems' compatibility issues, and the taxonomy will process the business intelligence needs of the provider.

As for every company these days, business intelligence becomes more and more important for logistics service providers too [15]. Logistics service providers can make use of business intelligence in many ways: using the logistics-specialized functions (carrier performance evaluation, mode-cost analysis, supplier compliance analysis, carrier relationship management, capacity planning, cycle time analysis, routing and scheduling, truck and driver performance analysis, root cause and claims analysis [13]), inventory and warehouse functions [18] (inventory analysis, warehouse performance analysis, assigning warehouse costs, picking analysis, warehouse space utilization analysis), functions in relation with added value (cost-benefit analysis, reverse logistics, assembly analysis, kitting), and of course the „usual” BI functions as well (supply chain visibility, forecasting and customized reports), with management information services

(dashboard reporting with KPI indicators), marketing and sales (customer service portfolio and customer profitability analysis with customer service level analysis), HR (reports, manpower allocation, training and succession planning with the usage of HR portals) and financial (budgetary analysis, fixed asset return analysis and financial ratio analysis) analytics services [12].

In the second part of the article handles the usual challenges in connection with business intelligence system introductory planning – concerning compatibility and connectivity issues – the critical points will be reviewed as well. The third part contains the results of the examination, with the focus on the available business intelligence systems, and their operating environment – taking into consideration their capabilities and given functionality. The last thought contains a well-known and approved taxonomy for BI (BI – business intelligence) systems, and a more modern and easier to understand taxonomy.

So, let us examine, how is it possible to adopt the most modern BI solutions into an environment with an already existing infrastructure – including hardware and software – and if it is possible to develop a usable taxonomy [16].

## **2. Business intelligence systems and their existing environment**

The assumption is current, because there are numerous studies on how to build cost effective corporate IT systems with integrated business intelligence solutions from scratch, but the majority of these studies do not focus on factors really typical to the corporate layer: the first factor is, that most companies do have a considerable past, and has to operate during the introduction process as well – or just very small halt is acceptable. The second factor is, that these corporate systems are very heterogeneous – on both hardware and software sides. The reason for this is in the evolutionary development of the companies, at many companies, the old systems are kept for a long time – either because it is needed, or there are no funds for replacing them.

At the planning / design stages of business intelligence system introductory projects those factors require extra attention – which often gives a crossroad opportunity for software engineers. The more expensive solution is always, if at the time of the introductory project refactoring and modernization gets place, with the re-thinking of the used solutions [1]. The more cost-effective solution is, however if we examine, if the already existing infrastructure could handle the extra load, and if so – would it be possible to place the BI system on top. If that is not the case, a wider range of possible solutions are needed to be examined.

Of course, the requirements, that the new BI system has to meet, have to be available at an early stage of the project, so the developer team can form the infrastructure in time [1].

If the soon-to-be-realized expectations are known, the resource needs can be calculated. As a general rule, the functions can be grouped as low resource requirement or high resource requirement.

The low resource requirement group contains typically periodical (so, continuous functionality or continuous calculation is not required, so the load waves can be dispersed smoothly) tasks, for example the daily statistical calculations.

The high resource requirement group contains typically continuously operating functions, or functions with real-time requirements. Business intelligence is usually seated in this group with other CRM and decision support solutions, since these functions require complex calculations. As the processable data size grows, so does the resource requirement of the operation.

In case, if there are no high resource requirement functions present in the specifications, it is verifiable through experience, that the already existing infrastructure will be able to handle the claims. Knowing the load waves (which can be determined at the early stages of the introductory projects situation analysis) so the regular functions executions can be timed not to be placed in a heavy load period – so the overload periods can be avoided.

In the probable case on the other hand, when there is a claim for functionality with high resource needs, it is necessary to think through the opportunities [2]:

- Rent infrastructure as a service: the full outsourcing of the IT infrastructure, known as the IaaS renting model.
- Renting platforms as a service: paying for the development platform (with the operating ones as well) with infrastructure – known as the PaaS service model.
- Using software as a service: the highest level service, many types are known, but the usage-dependent pricing with the fix pricing models are the most widespread.

These three levels of service can be perceived as a pyramid, where the most detailed service level is represented at the top – using software as a service, while at the bottom – with the lowest level – is the infrastructure as a service model [17].

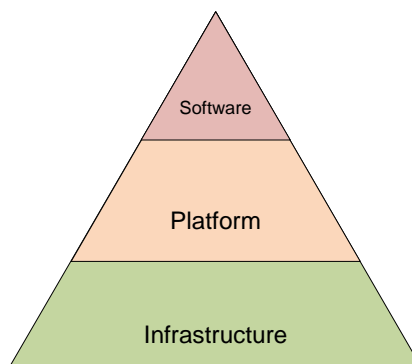


Figure 1. The usable service levels

It has to be decided in the early stages of the design phase, which service level to use (if any) – it is a good practice, to handle these levels as inputs for the Kesselring algorithm.

The number of business intelligence related software is around approximately around 60. (Of course, it does worth mentioning, that there are more complex software bundles, and there are simple systems with limited functionality – or specialized for only one task, for example data integration or data visualization.) Although that market has many competitors and of course there are less complex systems, it is hard to draw a line between these two categories.

The connection between the already existing infrastructure and the BI system has usually 2 critical points:

- The first one is the operating system (which handles the communication with the underlying hardware),
- And the second is the connection to the existing databases.

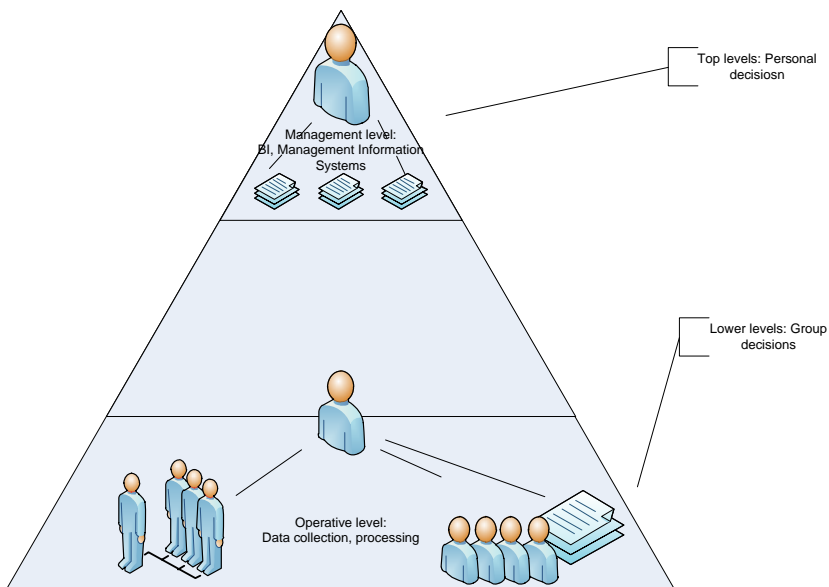


Figure 2. Place of BI in the company hierarchy

So, before putting the BI system into the infrastructure (or better said, on top of it – Fig.2.), it has to be taken into consideration, if it will be able to operate on the operating system, and use the existing databases. After the filtering, the remaining solutions have to be examined, on the functional side – if they are capable of meeting the requirements.

To decide whether it is possible, to integrate a BI solution into the existing infrastructure (or at least on platform and database sides), the available solutions have to be examined [17].

54 of the business intelligence solutions were processed in the examination. Most of these systems are commercially available, but there are free software, and open-source software between them.

### 3. Examination and results

The examination proved on hand, that the leaving the classical client-server architecture is in progress, aiming web architectures. On the other hand, it was revealed, that those systems built on the classical client-server architecture or simply desktop applications are using the widespread frameworks, like JAVA technology and .NET. The positive side of this is, that these frameworks are available on many of the used operating systems.

As for the operating systems on the already existing servers (if there are new hardware elements purchased along the way – at the introductory project, that will assign the whole project to a higher price range, but frees the project from dependencies) the elasticity of the products differ. Looking from this viewpoint, grouping upon, if more operating systems are supported looks like a good basis.

The bigger, more complex systems (especially those, with decades of development) thanks to their evolutionary development, can be said, that usually support the industry-standard operating systems on server side. These are:

- z/OS: The operating system for IBM mainframes. Supports backward compatibility, large memories and a series of mainframe technologies.
- Unix / Linux: Since the development of Unix in 1969, a series of unix-like operating systems came into use (Linus, BSD). The widespread available, free and open source solution was the Linux (and many versions are still), but has commercially available versions too.
- Solaris: A unix-like operating system developed by Sun Microsystems to support SPARC and x86 servers and workstations.
- Suse Linux: the oldest open source Linux version – there is a supported version for enterprises.
- Red Hat Enterprise Linux: the other enterprise Linux, having a considerable share.
- Windows: The Microsoft Windows had three versions on server side, which laid the foundations on the server side headway: Windows NT, Windows Server 2003 and Windows Server 2008 – the R2 release is also available. Typically used because .NET server side need, or IIS web server requirement.

The other group contains the less complex systems, providing less or a narrower range of functionality, capable only to solve a few, typical problems, and having less elasticity. These systems are generally not capable of server-side computing, and does not have server side functionality as well – so defined as desktop applications. Since it is easier to implement these systems with advanced developer tools, most of them were created using JAVA or the .NET framework. The benefit of a realisation like this is, that running the application will have only the prerequisite of installing the framework – those are available on most commercial operating systems.

While 25,9% of the examined solutions was written in JAVA language, 48,1% of these were developed to be run in windows environment. The two sets do not overlap, but worth mentioning, that all of the open source and all of the free systems were developed with JAVA.

The availabilities of JAVA environment are show in the following (Table 1.) table:

*Table 1. JAVA availability*

	32-bit version	64-bit version
Microsoft Windows 7	X	X
Microsoft Windows XP	X	X
Microsoft Windows Vista	X	X
Microsoft Windows Server 2003	X	X
Microsoft Windows Server 2008	X	X
Microsoft Windows 2000	X	
Sun Solaris	X	X
Linux	X	X
Apple OS X	Exist, own version.	

The .NET framework is available on the Microsoft operating systems (in certain versions it is preinstalled, and freely upgradable) and available on Linux as well, it is developed in the Mono project (Mac OS X and Solaris are supported too) – usually one step older version, than the newest available.

The screenshot displays the official Java website's download page, organized into four operating system categories: Windows, Solaris, Linux, and Apple (OS X). Each category lists specific Java versions with their file sizes and provides links for instructions and verification. A central warning message states: "We have detected you may be viewing this page in a 32-bit browser. If you use 32-bit and 64-bit browsers interchangeably, you will need to install both 32-bit and 64-bit Java in order to have the Java plug-in for both browsers." Below the Linux section, a note advises: "\* Please use the 32-bit version for Java applet and Java Web Start support." At the bottom, a note for Apple users states: "Apple supplies their own version of Java. Use the Software Update feature (available on the Apple menu) to check that you have the most up-to-date version of Java for your Mac."

Operating System	Version	File Size	Instructions	Verify Now
Windows	Windows 7/XP/Vista/2000/2003/2008 Offline (32-bit)	15.3 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
	Windows 7/XP/Vista/2003/2008 (64-bit)	15.4 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
Solaris	Solaris (32-bit)	24.8 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
	Solaris (64-bit)*	10.6 MB + 24.3 MB 32-bit Solaris	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
	Solaris x86	18.9 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
	Solaris x64 *	6.9 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
Linux	Linux RPM (self-extracting file)	19.7 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
	Linux (self-extracting file)	20.2 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
	Linux x64 *	19.6 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
	Linux x64 RPM *	189.1 MB	<a href="#">Instructions</a>	<a href="#">Verify Now</a>
Apple (OS X)	Apple supplies their own version of Java. Use the Software Update feature (available on the Apple menu) to check that you have the most up-to-date version of Java for your Mac.			

Figure 3. JAVA availabilities from the official JAVA page.

The question – whether (which is less expensive) to use free software with more expensive support and staff or use commercial software, with cheaper support and staff – often induces heated debate on professional (and on-line) forums.

The other important technological factor is the database connection. It had to be examined, what database connectivity capabilities do the available business intelligence solutions have. Of course, in this aspect, there are many ways, if the native support is no available: for example third-party ETL software, or mid-tier applications. During the examination the following database connection opportunities were covered:

Working with developer-specific / product-specific data sources: specially the feature for large software bundles, from developer companies, which are capable to give the full technology stack needed for enterprises. These products a trying to drive their clients to integrate more and more of their solutions, by preferring these data sources. Approximately 44,4% of the examined solutions are capable of using product-specific data sources.

SQL: the simple query language, known by almost every database-related software [5]. However there are product-specific differences from the standard, like Ms-SQL or the PL/SQL used by Oracle. All of the examined solutions were capable of using SQL, but worth mentioning that in some cases, because of a different programming language, those commands needed to be wrapped in a way.

XML: the text format of W3C, specially designed to deliver internet documents. It does support Unicode characters, usable in many programming languages, and by producing web services or data sources [6]. The usage requires a special API, but that is available in all languages – 72% of the examined solutions were capable of using XML.

ODBC: Open DataBase Connectivity, an interface developed to be independent from programming languages and operating systems [7]. With its help, the knowledge of the ODBC-defined interface is enough; and from then it is the duty of the driver present, to translate the commands to the database management system [8] – 54% of the examined solutions were capable of using ODBC data sources.

JDBC: Java DataBase Connectivity, an API to JAVA language, to reach relational databases [9]. With the help of the JDBC-ODBC bridge, Java can access the original ODBC functionality. The equivalent for .NET environment is the ADO.NET [10]. Naturally, every single solution written in JAVA was able to use JDBC data sources.

Thanks to the application of these standards, the business intelligence systems are capable of using the optional databases. There are exceptions of course, and in these cases it is expedient to look for other connection opportunities – if necessary.

## **4. Goals and importance of the taxonomy**

Taxonomy is a hierarchical classification or ordering method, which can be done at any number of dimensions.

A very important expectation to any taxonomy is, that every element of the described space has to have a place in it, so in this case: we have to be able to place every single business intelligence application (already existing or future development).

If every solution has a place in a taxonomy, then it can be used at the inspection of alternatives, at a system introductory process – or in other words: an advice can be made based on limited number of information, shrinking the circle of available systems, so it will be easier to choose the right solution.

### **4.1. The origin of business intelligence taxonomy**

Since business intelligence systems are one of the most popular IT systems of the decade, the number of publications in connection grew great – which turns out to be much less, if we do not count the marketing aimed publications.

The most cited and still valid taxonomy was created by Colin White. (He is a leading researcher in BI, a speaker at many significant conferences, and he is in connection with <http://bi-research.com> and <http://www.b-eye-network.com> web portals – he has periodical publications in the topic.) That (first) taxonomy was done in 2004. The base of the taxonomy was to divide the business intelligence applications (which only cover a part of the whole BI process) from business intelligence platforms (which can provide a base for building integrated, more sophisticated, company-wide systems).



According to the taxonomy, business intelligence applications can be grouped in many ways. The first of these is the timeframe, and the decision level, which they support: so strategic decisions (of creating strategic plans), tactical or operational decisions. These systems can be grouped according to their granularity: the outputs can be detailed or summarized. The next grouping rule is the type of the used data store: data warehouses, data marts, OLAP cubes, immediately accessible stores, data caches, snapshots and virtual views. The next dimension is the applicable business rules: these can be transformations, analyses, using metrics, creating actions, handling exceptions and advices, which action to take [11].

Every type of business intelligence can have different timeness or latency, so the diversification follows: there are real-time, near real time and historical systems. The next dimension is the type: there are applications capable of handling transactional data, using business / performance / special field indicators, and those, which are able to automate warnings, actions and offers.

The more complex systems, which are made up many integrated and cooperating applications were already sold in packages; so that is the next dimension: there are developer packages, data integration packages, coordinating, designer and forecasting packages.

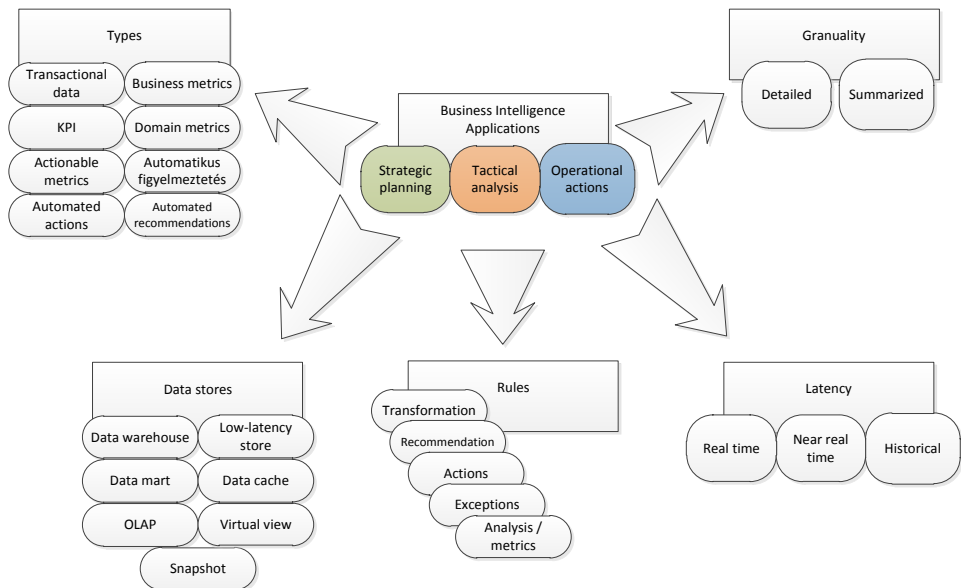


Figure 5. An already existing taxonomy for business intelligence applications

The developer package is defined by the dashboard-making application, some kind of interactive development environment, rule engine, analytical application, a reporting studio and performance management application. A data integration package is defined by the data modeller and exploratory applications, the ETL designer with metadata management capabilities, considering data quality as well. A designer and forecasting package usually contains data mining and text-mining solutions, cost management and

forecasting applications. A system introductory package and a teamwork promotional package is described by CMS, dashboard, portal and teamwork enabling services. A bit more divided, but the last dimension is the place, where these systems are used: front-office, middle-office or back-office [11]. Aside from this, during the last decade a respectable amount of categorisation was created.

#### **4.2. New order – new taxonomy**

The mentioned and described taxonomy is considered good, but the 8 years, since it was created contained many new business intelligence solutions, and the market competition is ever-sharpening in this sector. Beside that, many new systems can cover more areas, and reordering of the market caused the blurring of borders in these categories. But the greatest reason for making the taxonomy more simple is the causal relation created by market trends (the price of technology dropping).

To create a simplified taxonomy it is needed to reduce of dimensions in the model. The easiest way doing this is looking for relationships between dimensions. During the search, 3 methods were used: the mentioned BI research, interviews conducted with professionals and years of practice.

According to the results, the granularity of the results is not necessary, since almost every system has settings to provide detailed or summarized results. Keeping the timeframe is useful on the other hand, to diversify between strategic planning, tactical and operational decisions.

Keeping the latency in also proved useful, although the examination and the interviews showed, that there is a weak connection between latency and timeframe: in most cases, a product capable of strategic planning, can also use historical data, and products with tactical decision support are capable of using near real-time data, and of course the operational decision supporting systems usually capable to use real-time data. However, these categories should not taken strictly.

In my opinion, the third dimension should contain the development capabilities of a given product (Fig. 7.). So, in this dimension the following categories are expected: COTS products (Commercially-Off-The-Shelf, or commercially available) – these products can be described with easy installation process and limited settings with simple functionality, Customizable COTS products (products with a wide range of settings and connection points, but no development environment in IT sense), COTS products with development suite (containing a development environment, usually containing a programming language) and in-house developments (specially designed to solve a specific problem).

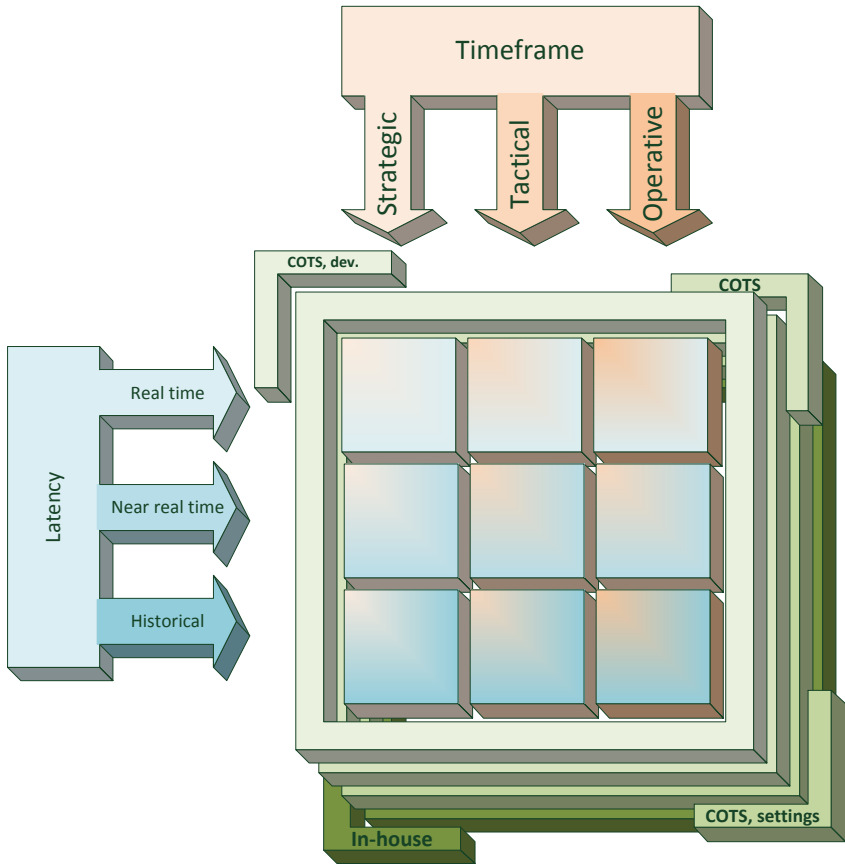


Figure 6. The new taxonomy

Since the taxonomy has to be capable of classification of every single BI solution; it has to be tested. Let the example be the SAS Enterprise Guide application (Fig. 7.). This software is capable mostly of strategic planning, and less of tactical decision support – since it is a management information system; mostly capable of using historical data, but the possibility is given of using near real-time data as well. Itself is a COTS product, with a wide range of customization available. It is a different question, that the developer of this product (SAS) offers a whole development environment, and a wide range of BI support products as well – so given the chance, and buying the whole package would cover the entire space in the taxonomy.

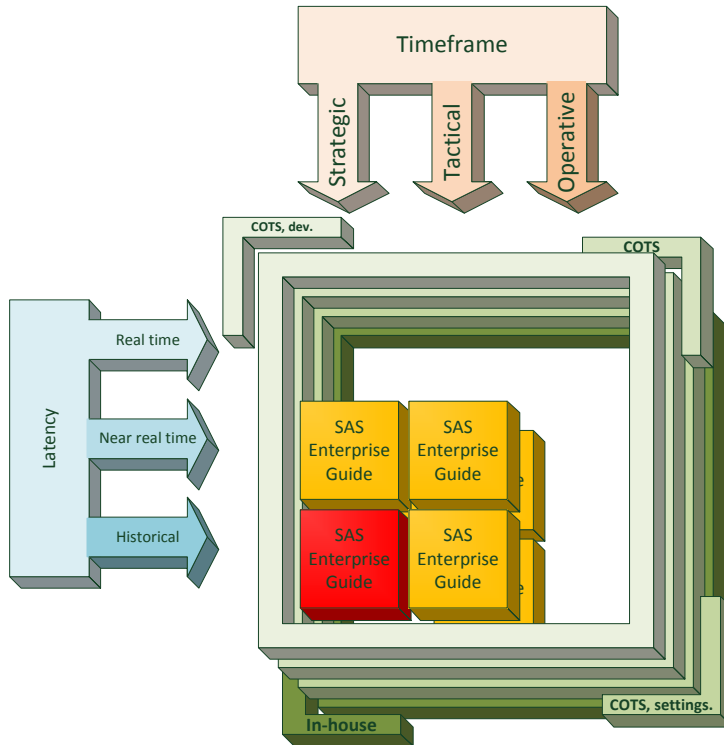


Figure 7. SAS Enterprise Guide placed in taxonomy

All 54 of the BI applications can be placed in a same way into the taxonomy. During the interviews, the question arose, if the interactive publication of the whole survey would help (or would make easier) at the beginning of a system introductory project. Nearly 70% of the answers were solid yes' with the constraint, that even with the taxonomy, there are many solutions with similar functionality, so feasibility studies (the second part of the article contains the compatibility issues) and introductory plans are needed, to clarify, how can those solutions serve the company's needs.

## 5. Conclusion

At last, it can be said – without citing the need for BI systems, or their efficiency – that it is a great challenge to implement a business intelligence solution into a existing infrastructure.

It is not easy to conquer the challenge: the same preparation is needed, as any other software project (even a specialized methodology exists, proportioning the task to steps) and the same amount of attention – if not more – is needed. One of the initial steps is choosing the applicable software components: it is important, that it has to be compatible with the already running (existing) systems, databases, infrastructure and data sources.

## References

- [1] Moss, L. T., Atre, S.: *Business intelligence roadmap: the complete project lifecycle for decision-support applications*, Addison-Wesley, 2003
- [2] Velte, T., Velte, A., Velte, T. J., Elsenpeter, R. C.: *Cloud Computing: A Practical Approach*, McGraw Hill Professional, 2009
- [3] Castellanos, M., Dayal, U., Sellis, T.: *Business Intelligence for the Real-Time Enterprise*, Springer, 2009
- [4] Jarke, M.: *Fundamentals of data warehouses*, Springer, 2003
- [5] Opper, A. J.: *SQL demystified*, McGraw-Hill/Osborne, 2005
- [6] Harold, E. R., Means, W. S.: *XML in a nutshell*, O'Reilly Media, Inc., 2004
- [7] Geiger, K.: *Inside ODBC*, Microsoft Press, 1995
- [8] SAS Institute: *SAS/Access 9.1 Supplement for ODBC (SAS/Access for Relational Databases)*, SAS Publishing, 2004
- [9] Van Haecke, B.: *JDBC 3.0: Java database connectivity*, John Wiley & Sons, 2002
- [10] Otey, M., Otey, D.: *ADO.NET: the complete reference*, McGraw-Hill/Osborne, 2003
- [11] White, C.: *A taxonomy for BI*, Information-management.com, 2004
- [12] Rao P, S., Swarup, S.: *Business intelligence and logistics*, Wipro Technologies, 2001
- [13] Vercellis, C.: *Business Intelligence: Data Mining and Optimization for Decision Making*, John Wiley and Sons, 2009
- [14] Thierauf, R. J.: *Effective business intelligence systems*, Greenwood Publishing Group, 2001
- [15] Sabherwal, R., Becerra-Fernandez, I.: *Business Intelligence (Practices, Technologies and Management)*, John Wiley and Sons, 2010
- [16] Rasmussen, N., Goldy, P. S., Solli, P. O.: *Financial business intelligence: trends, technology, software selection, and implementation*, John Wiley and Sons, 2002
- [17] Page, J.: *Building a business intelligence architecture fit for the 21st century*, Melrose Books, 2009
- [18] Reynolds, J.: *Logistics and fulfilment for e-business: a practical guide to mastering back office functions for online commerce*, Focal Press, 2001